

# ECE 530: CLOUD COMPUTING

## KAFKA READING ASSIGNMENT

DAVID KIRBY – 101652098 – DAVIDKIRBY@UNM.EDU

SPRING 2021

In this paper, the authors describe their design and implementation of a distributed messaging system for collecting and delivering log data with minimal latency. The authors state that previous solutions for this sort of data meant physically scraping log files for analysis. More modern systems have implemented aggregators to automate the scraping process for offline processing, dumped in a data warehouse. What the author are proposing in their messaging system, Kafka, allows for support of real-time applications logging and processing in addition to traditional offline analytics, even providing an API to allow other applications to consume this real-time data.

To create Kafka, the authors focused on three design principles: efficiency on a single partition, distributed coordination, and delivery guarantees. In experiments, the authors show that, when compared with other messaging services (e.g., Apache ActiveMQ and RabbitMQ), Kafka achieves much higher throughput. It also provides distributed support and scales incredibly well. After having implemented a version of RabbitMQ when we built our OpenStack deployment for homework #1, I am curious to research more into Kafka and its other implementations.